

AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented process for reordering items in a database to be retrieved for display to a user, comprising the steps of:  
ordering storing a plurality of words stored in a linguistic database (LDB), said plurality of words ordered in said LDB according to a predefined linguistics frequency of use model;  
associating each word stored in said LDB with an object number;  
accepting with a processor user input from a keyboard, said user input comprising at least one keypress;  
retrieving any words from said LDB that matches at least one letter corresponding to said at least one keypress match said user's input;  
displaying a list of said retrieved words as ordered in said LDB;  
if said list does not include a word, accepting said a user-defined word as defined input by said user;  
storing said user-defined word in a user database (UDB);  
assigning a frequency count to said user-defined word;  
storing said frequency count for said user-defined word in said UDB;  
enabling said user to select a word from said displayed list;  
assigning said a frequency count to every selected word in a non first order position in said displayed list and a frequency count to a first order word, if a non first order word is selected, said frequency count for said non first order selected word being different than said frequency count for said first order non selected word;  
storing said frequency count of said each selected word and said word's corresponding object number in association with each word in said UDB;  
accepting a subsequent user input from a keyboard, said user input comprising at least one keypress;  
retrieving any words from said LDB and any user-defined words from said UDB that match at least one letter corresponding to said user's subsequent input at least one keypress;  
if more than one word from any of said LDB and said UDB is retrieved matches said user's subsequent input and at least one retrieved word is associated with a frequency count, dynamically reordering for display said retrieved any matching words in said LDB as ordered in said LDB and any matching words in said UDB according to said

frequency count as a function of said predefined linguistics frequency of use model and each frequency count associated with any of said retrieved words; and

displaying a list of said reordered matching words if more than one word matches said user's subsequent input; and

updating a frequency count associated with a word each time a non first order word is selected from a displayed list of matching words.

2.-4. (Cancelled)

5. (Currently Amended) The process of Claim [[4]] 1, wherein if said a non first ordered word is selected by said user a predetermined number of times then said non first ordered word is displayed in a first ordered position in said list of reordered matching words.

6. (Currently Amended) The process of Claim [[4]] 1, wherein all non first ordered words that are assigned said frequency count are initially assigned equal frequency counts.

7. (Previously Presented) The process of Claim 1, wherein said word's frequency count is increased each time said user selects said word.

8. (Previously Presented) The process of Claim 1, wherein if said user selects said word from said list of retrieved words or said list of reordered words that is below a second ordered position, then said selected word is assigned said frequency count that places said word in said second ordered position in said list.

9. (Currently Amended) A computer-implemented process for reordering items in a database to be retrieved for display to a user, comprising the steps of:

ordering storing a plurality of words stored in a linguistic database (LDB), said plurality of words ordered in said LDB according to a predefined linguistics frequency of use model;

associating each word stored in said LDB with an object number;

accepting with a processor user input from a keyboard, said user input comprising at least one keypress;

retrieving any words from said LDB that match at least one letter corresponding to said user's input at least one keypress;

displaying a list of said retrieved words as ordered in said LDB;

~~if said list does not include a word, accepting said a user-defined word as defined input by said user;~~

storing said user-defined word in a user database (UDB);

assigning a frequency count to said user-defined word;

storing said frequency count for said user-defined word in said UDB;

enabling said user to select a word from said displayed list;

assigning said a frequency count to every selected word in a non first order position in said displayed list and a frequency count to a first order word, if a non first order word is selected, said frequency count for said first order non selected word;

storing said frequency count of said each selected word and said word's corresponding object number in association with each word in said UDB;

storing a frequency count for a first order word from said displayed list if a non first order word is selected from said displayed list, said frequency count for said non selected first order word being lower than said frequency count for said selected non first order word, said frequency count being stored in association with said first order non selected word in said UDB

accepting a subsequent user input from a keyboard, said user input comprising at least one keypress;

retrieving any words from said LDB and any user-defined words from said UDB that match at least one letter corresponding to said user's subsequent input at least one keypress;

if more than one word from any of said LDB and said UDB is retrieved matches said user's subsequent input and at least one retrieved word is associated with a frequency count, dynamically reordering for display said retrieved any matching words in said LDB as ordered in said LDB and any matching words in said UDB according to said as a function of said predefined linguistics frequency of use model and each frequency count associated with any of said retrieved words;

displaying a list of said reordered matching words if more than one word matches said user's subsequent input;

updating a frequency count associated with a word each time a non first order word is selected from a displayed list of matching words; and

periodically aging said frequency counts in said reorder database by reducing said frequency counts by a predetermined factor.

10. (Previously Presented) The process of Claim 1, further comprising the step of: periodically checking free space in said UDB and, if said free space is less than a predetermined amount, then removing from said UDB said words that have said frequency counts below a predetermined threshold.
11. (Previously Presented) The process of Claim 10, wherein said removing step removes said user-defined words having said frequency counts below said predetermined threshold after removing other words having said frequency counts below said predetermined threshold.
12. (Previously Presented) The process of Claim 1, further comprising the step of: resolving frequency collisions when two words in said list of reordered words are associated with equal frequency counts by ordering said word that was most recently selected first.
13. (Previously Presented) The process of Claim 1, further comprising the step of: resolving frequency collisions when two words in said list of reordered words are associated with equal frequency counts by ordering said word having a higher initial ordering in said LDB first.
14. (Previously Presented) The process of Claim 1, further comprising the step of: resolving frequency collisions in said list of reordered words when a user-defined word and a word from said LDB have equal reordering frequency counts by ordering said user-defined word first.
15. (Currently Amended) A computer-implemented process for reordering items in a database to be retrieved for display to a user, comprising the steps of:  
ordering storing a plurality of words stored in a linguistic database (LDB), said plurality of words ordered in said LDB according to a predefined linguistics frequency of use model;  
~~associating each word stored in said LDB with an object number;~~

accepting with a processor user input from a keyboard, said user input comprising at least one keypress;

retrieving any words from said LDB that matches at least one letter corresponding to said at least one keypress match said user's input;

displaying a list of said retrieved words as ordered in said LDB;

if said list does not include a word, accepting said a user-defined word as defined input by said user;

storing said user-defined word in a user database (UDB);

assigning a frequency count to said user-defined word;

storing said frequency count for said user-defined word in said UDB;

enabling said user to select a word from said displayed list;

assigning said a frequency count to every selected word in a non first order position in said displayed list and a frequency count to a first order word, if a non first order word is selected, said frequency count for said non first order selected word being different than said frequency count for said first order non selected word;

storing said frequency count of said each selected word and said word's corresponding object number in association with each word in said UDB;

accepting a subsequent user input from a keyboard, said user input comprising at least one keypress;

retrieving any words from said LDB and any user-defined words from said UDB that match at least one letter corresponding to said user's subsequent input at least one keypress;

if more than one word from any of said LDB and said UDB is retrieved matches said user's subsequent input and at least one retrieved word is associated with a frequency count, dynamically reordering for display said retrieved any matching words in said LDB as ordered by said LDB and any matching words in said UDB according to said as a function of said predefined linguistics frequency of use model and each frequency count associated with any of said retrieved words; and

displaying a list of said reordered matching words if more than one word matches said user's subsequent input;

updating a frequency count associated with a word each time a non first order word is selected from a displayed list of matching words; and

wherein words selected by said user that do not have a possibility of a collision with other words are not assigned said a frequency count.

16. (Currently Amended) An apparatus for reordering items in a database to be retrieved for display to a user, comprising:

a module for accepting user input from a keyboard, said user input comprising at least one keypress;

a linguistic database (LDB) containing a plurality of words ordered according to a predefined linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match at least one letter corresponding to said at least one keypress said user input, said words retrieved from any of said LDB and from said UDB;

said UDB for storing any user-defined words entered by said user, a frequency count associated with each for said user-defined word[[s]], and a frequency count associated with each word stored in said LDB that was assigned a frequency count by an assigning module and any word selected by said user from said list of any words that match said user input, and an object number corresponding to said frequency count for any word selected by said user;

a module for retrieving from any of said LDB and from said UDB a list of any words that match at least one letter corresponding to said at least one keypress of said user's input, said words dynamically reordered for display of said retrieved words as a function of according to said predefined linguistics frequency of use model and each frequency count associated with any of said retrieved words; and

said assigning module for assigning said a frequency count to any every selected word[[s]] in a non first order position in a list of said retrieved words and assigning a frequency count to a first order word if a word in a non first order position is selected, said frequency count being different for said first order word than said frequency count for said selected non first order word, said assigning module updating a frequency count each time a non first order word is selected from said retrieved list selected by said user and inserting said selected words' frequency counts into said UDB.

17.-18. (Canceled)

19. (Previously Presented) The apparatus of Claim 16, wherein said frequency count is assigned to said selected word if said selected word is in a non first order position in said list and is selected for a first time.

20. (Previously Presented) The apparatus of Claim 19, wherein if said non first ordered word is selected by said user a predetermined number of times then said non first ordered word is displayed in a first ordered position in said list.

21. (Previously Presented) The apparatus of Claim 19, wherein all non first ordered words that are assigned said frequency count are initially assigned equal frequency counts.

22. (Previously Presented) The apparatus of Claim 16, wherein said word's frequency count is increased each time said user selects said word.

23. (Previously Presented) The apparatus of Claim 16, wherein if said user selects from said list a word that is below a second ordered position then said assigning module assigns a frequency count that places said word in said second ordered position in said list.

24. (Currently Amended) An apparatus for reordering items in a database to be retrieved for display to a user, comprising:

a module for accepting user input from a keyboard, said user input comprising at least one keypress;

a linguistic database (LDB) containing a plurality of words ordered according to a predefined linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match at least one letter corresponding to said at least one keypress said user input, said words retrieved from any of said LDB and from said UDB;

said UDB for storing any user-defined words entered by said user, a frequency count associated with each for said user-defined word[[s]], and a frequency count associated with each word stored in said LDB that was assigned a frequency count by an assigning module and any word selected by said user from said list of any words that

~~match said user input, and an object number corresponding to said frequency count for any word selected by said user;~~

a module for retrieving from any of said LDB and from said UDB a list of any words that match at least one letter corresponding to said at least one keypress of said user's input, said words dynamically reordered for display of said retrieved words as a function of according to said predefined linquistics frequency of use model and each frequency count associated with any of said retrieved words;

said assigning module for assigning said a frequency count to every selected word[[s]] in a non first order position in a list of said retrieved words and assigning a frequency count to a first order word if a word in a non first order position is selected, said frequency count being different for said first order word than said frequency count for said selected non first order word, said assigning module updating a frequency count each time a non first order word is selected from said retrieved list selected by said user and inserting said selected words' frequency count into said UDB; and

a module for periodically aging said frequency counts in said reorder database by reducing said frequency counts by a predetermined factor

25. (Previously Presented) The apparatus of Claim 16, further comprising:

a module for periodically checking for free space in said UDB and, if said free space is less than a predetermined amount, then removing from said UDB said frequency counts and corresponding object numbers or corresponding user-defined words for words that have frequency counts below a predetermined threshold.

26. (Previously Presented) The apparatus of Claim 25, wherein said removing module removes user-defined words having frequency counts below said predetermined threshold after removing other words having frequency counts below said predetermined threshold.

27. (Previously Presented) The apparatus of Claim 16, further comprising:

a module for resolving frequency collisions when two words in said list have equal frequency counts by ordering said most recently selected word first.

28. (Previously Presented) The apparatus of Claim 16, further comprising:

a module for resolving frequency collisions when two words in said list have equal frequency counts by ordering said word having a higher initial ordering in said LDB first.

29. (Previously Presented) The apparatus of Claim 16, further comprising:  
a module for resolving frequency collisions in said list when said user-defined word and said word from said LDB have equal frequency counts by ordering said user-defined word first.

30. (Currently Amended) An apparatus for reordering items in a database to be retrieved for display to a user, comprising:  
a module for accepting user input from a keyboard, said user input comprising at least one keypress;

a linguistic database (LDB) containing a plurality of words ordered according to a predefined linguistic frequency of use model;

a module for displaying to said user a list of any words in said LDB and any user-defined words in a user database (UDB) that match at least one letter corresponding to said at least one keypress said user input, said words retrieved from any of said LDB and from said UDB;

said UDB for storing any user-defined words entered by said user, a frequency count associated with each for said user-defined word[[s]], and a frequency count associated with each word stored in said LDB that was assigned a frequency count by an assigning module and any word selected by said user from said list of any words that match said user input, and an object number corresponding to said frequency count for any word selected by said user;

a module for retrieving from any of said LDB and from said UDB a list of any words that match at least one letter corresponding to said at least one keypress of said user's input, said words dynamically reordered for display of said retrieved words as a function of according to said predefined linguistics frequency of use model and each frequency count associated with any of said retrieved words; and

said assigning module for assigning said a frequency count to every selected word[[s]] in a non first order position in a list of said retrieved words and assigning a frequency count to a first order word if a word in a non first order position is selected, said frequency count being different for said first order word than said frequency count

for said selected non first order word, said assigning module updating a frequency count each time a non first order word is selected from said retrieved list selected by said user and inserting said selected words' frequency count into said UDB;

wherein said words selected by said user that do not have a possibility of a collision with other words are not assigned said frequency count.

31. (Previously Presented) The process of Claim 1 wherein when said word from said LDB is selected for a first time, said step of assigning said frequency count generates said frequency count from said word's position in said LDB as a function of said predefined linguistics frequency of use model.

32.-33. (Canceled)

34. (Previously Presented) The process of Claim 31, wherein said user-defined words are initially assigned equal frequency counts during said assigning step.

35.-44. (Canceled)

45. (Previously Presented) The apparatus of Claim 16 wherein said assigning module assigns said frequency count for said word from said LDB based on said predefined linguistics frequency of use model.

46.-47. (Canceled)

48. (Currently Amended) The apparatus of Claim 45, wherein said user-defined words are initially assigned equal frequency counts by said assigning module.

49.-60. (Canceled)